Email Based Communication On MasMT: Enables Human Agent Interaction

JAP Madhubhani[1], B Hettige[2]

[1] Department of Computer Engineering, General Sir John Kotelawala Defence University
[2] Department of Knowledge Engineering and Communication, University of Sri Jayewardenepura
Author email: 39-bce-0023@kdu.ac.lk

DOI: 10.29322/IJSRP.16.01.2026.p16919
https://dx.doi.org/10.29322/IJSRP.16.01.2026.p16919

Paper Received Date:15th December 2025
Paper Acceptance Date: 12th January 2026
Paper Publication Date: 20th January 2026

Abstract- Human-Agent Interaction (HAI) studies the methods through which people work together with intelligent agents. The research introduces a new approach to merge email communication with MaSMT which stands as a lightweight multi-agent system framework to support email-based human-agent interactions. The system enables human users to function as agents through email which serves as an accessible asynchronous communication channel that most multi-agent frameworks lack. The paper explains why email functions as a communication interface for MaSMT and provides background information about multi-agent systems and HAI principles including engagement and intentionality and scenario-based interaction. The paper explains the system design of email-based MaSMT integration through its main elements and interaction processes. The paper demonstrates email-based agent communication through a classroom reservation scenario to show both advantages and difficulties of this method. The research shows that email interfaces make agent interactions more human-friendly by allowing people to guide automated systems yet reveals problems with unclear messages and delayed responses. The communication system allows users to experience more natural human-agent interactions through its improved accessibility and better intention recognition of user intentions and flexible scenario handling. The research establishes future directions for multi-agent system development while demonstrating how this communication model enhances human-agent interaction.

Index Terms- *Email-based Communication, Human-Agent Interaction, MaSMT Framework, Multi-Agent Systems*

I. INTRODUCTION

The integration of intelligent systems into daily life has created a situation where people struggle to distinguish between human choices and automated machine operations. Human-Agent Interaction (HAI) represents a fundamental artificial intelligence field which studies human interactions with intelligent agents. The study of HAI serves two essential purposes because it enables process automation and enables developers to create systems which reflect human values and behavioral patterns. Traditional multi-agent systems (MAS) operate as self-governing software agents which function independently in digital environments through negotiation and reasoning and minimal human supervision. The system model works well for various computational operations yet prevents human participation during operations that require human input or situational awareness or ethical decision-making. The conventional MAS architectures fail to incorporate built-in mechanisms which enable human users to actively join agent dialogues and decision-making processes.

MaSMT represents a Java-based framework which enables developers to build message-driven multi-agent systems through its lightweight design. Hettige et al. first presented MaSMT as a framework for English–Sinhala machine translation before its development[1]. The agent architecture of MaSMT consists of manager agents which oversee ordinary agents (client agents) that operate in swarms. Manager agents oversee groups of ordinary agents while using rule-based ontologies to distribute tasks and maintain message queue communication between agents. Ordinary agents perform particular tasks through their local message queues which connect to their manager and other agents in their swarm. The FIPA-ACL standard governs MaSMT message structure which includes sender and receiver information along with content and ontology and protocol fields. The structured message-based architecture of MaSMT enables parallel execution and dynamic agent interactions which result in better performance than rule-based systems in machine translation applications.

The implementation of human input into MaSMT faces multiple obstacles despite its flexible design and successful operation in

software-only environments[2]. The current HAI research focuses on developing AI systems which allow agents to request human input for decision-making when needed. A scheduling agent requires human input to resolve conflicts because human preferences and contextual understanding enhance decision quality. Most current MAS systems fail to provide users with simple methods to participate without requiring advanced interfaces or technical knowledge. The research introduces email communication as a solution to enable human users to interact with agents operating within the MaSMT framework. The universal accessibility of email combined with its asynchronous nature makes it an ideal tool for agent interaction because users can access it through their everyday email accounts.

Through email-based messaging in MaSMT users can function as online agents who send commands and receive responses from agents through their familiar communication platform. The system enables regular users to actively participate in MAS operations through a user-friendly interface that eliminates the need for technical knowledge of agent systems.

The main reason for adding email functionality to MaSMT is to create more human-friendly multi-agent system capabilities. The system uses email functionality to merge human knowledge with automated systems for developing flexible intelligent systems that operate effectively in actual environments. The research presents main contributions to the field: first MaSMT system extension through email communication for HAI, Second evaluation of the proposed method against existing HAI and MAS research findings, Third demonstration of the system through a practical application, and Fourth evaluation of email-based human-agent communication advantages and disadvantages.

The following sections present the research findings. The second section examines human-agent interaction systems and scenario-based multi-agent platforms to establish the foundation for our proposed solution. The third section explains the approach for adding email functionality to MaSMT through its system design and implementation process. The fourth section demonstrates an email-based classroom reservation system as a practical example to show how email-based HAI improves user engagement and decision flexibility and intentionality. The final section presents research findings and discusses system expansion possibilities and HAI system enhancements for multi-agent systems.

II. Literature Review

Human-Agent Interaction (HAI) studies the complex process of how intelligent agents share control and communicate with human users. Research indicates that systems need to integrate human input with machine autonomy in a meaningful way. Mostafa et al. (2016) developed a flexible HAI system which enables agents to change their autonomy level based on human supervisor needs[3]. The system allows human intervention during critical situations and normative decision-making through its adjustable autonomy feature which matches our email-based system that uses email to request human assistance. The system requires flexibility because it operates in unmanaged and open environments where pre-defined agent behaviors fail to handle unexpected situations.

A. Adjustable Autonomy in Human–Agent Interaction

Human–Agent Interaction (HAI) studies the methods through which intelligent agents enable human interaction “Fig. 1” and collaborative control. Research indicates that systems should unite human input with machine autonomy instead of implementing complete automated decision systems. Mostafa et al. develop a HAI framework which enables agents to modify their operational autonomy based on human supervisor needs. The system allows human operators to take control of agent operations during critical situations that require their intervention[3]. The system enables agents to stop their automated work through email before they ask for human assistance before continuing their process. The system requires flexibility because it operates in unmanaged open environments where pre-defined behaviors fail to handle unexpected situations.

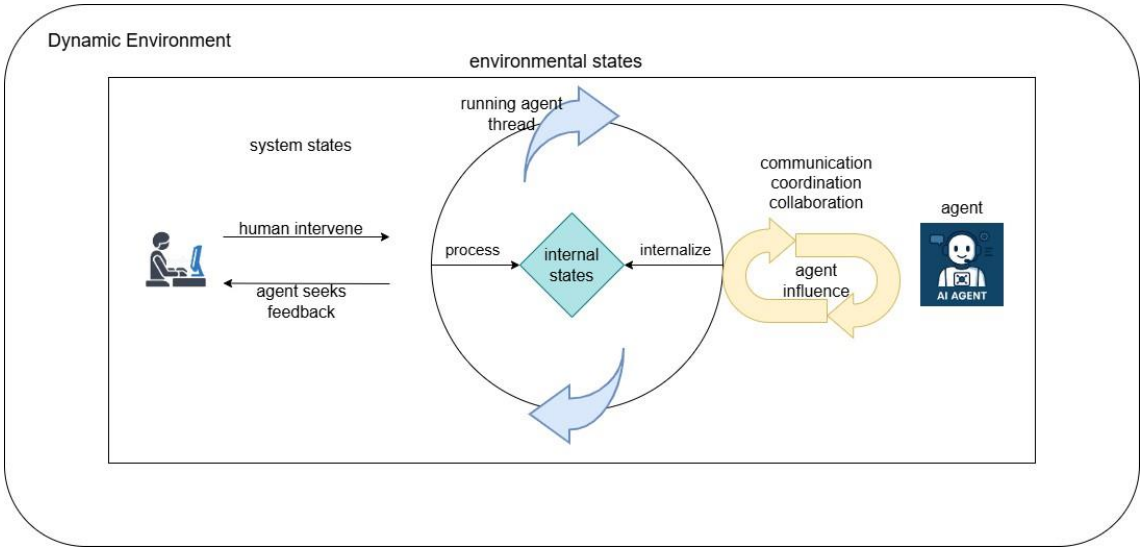

Figure 1: Human Intervention and Agent Influence within a Dynamic Environment

B. Conversational Agents and Asynchronous Interaction

The development of conversational agents and chatbots has accelerated because of NLP and deep learning progress which brought transformer models like BERT and GPT into existence. Kusal et al. (2022) conducted a scoping review of AI-based conversational agents to demonstrate how these systems have achieved major improvements in language processing and dialogue system management[4]. Most current interfaces operate under the assumption that users will interact in real-time through instant messaging or voice channels. Ahmed et al. (2024) explain that deep-learning-based HAI agents fail to achieve true context-awareness and maintain continuous conversations[5]. The system performance deteriorates when users interact through asynchronous channels because these channels do not support continuous real-time communication.

Our research project uses email as a communication platform to fill the existing gap in communication systems. The email system operates through a low-bandwidth asynchronous system which enables users to send detailed messages that agents can answer with extended explanations. Users can create elaborate requests through the system while agents provide detailed responses to their inquiries. The system accepts delayed responses because it provides better clarity and maintains audit trails and keeps records of all interactions (e.g. scheduling and collaborative planning). The research field lacks studies about asynchronous HAI which makes our work an original contribution to the field.

C. Scenario-Based Frameworks for HAI

Research studies employ scenario-based frameworks to develop HAI systems. The agent scenario mechanism (FSDL – a visual scripting language) developed by Chin et al. enables users without programming skills to create interaction scenarios through structured template-based design[6]. The method enables users to create pre-defined scripts which define agent actions for particular use cases while enabling human-agent interaction in regulated environments. The scenario-based design approach enables developers to create simple systems for established situations but it restricts flexibility when dealing with unexpected situations. The agents execute their programmed scripts but they become ineffective when operating beyond their predetermined paths.

The email-based interaction system enables users to send unrestricted natural language emails for requesting system actions. The system needs to understand email content instead of using pre-defined scripts because this approach improves scenario flexibility. The MaSMT infrastructure supports various domains and task types through email parsing rules and NLP models that convert user input into internal agent operations. The system's ability to adapt to user needs becomes possible through this flexible design approach which moves complexity from scenario development to interpretation and understanding processes.

D. Engagement in Human–Agent Interaction

The process of human integration into MAS systems creates challenges regarding user participation because users need to stay active and responsive throughout the entire process. Oertel et al. (2020) studied HAI engagement through real-time multimodal cue analysis of user gaze and prosody and hand gestures[7].

The way users engage with email-based systems operates through distinct mechanisms. Users control the pace of their interactions through periodic exchanges of messages.

- Users lack the ability to send immediate nonverbal signals such as head nods or "mm-hmm" responses.
- Users must wait for their turn in the conversation because it happens at different times.

Users can respond to messages whenever they choose because email threads maintain continuous conversation flow through threaded messages. The ongoing message history in email threads solves the problem of maintaining dialogue context which affects extended conversations. Users tend to create better human-agent collaborations because they feel less rushed to send well-prepared requests and responses through email.

E. Intentionality and the Intentional Stance

The dimension of HAI includes intentionality which refers to how users view the agent as having its own mental processes. The concept of Dennett's Intentional Stance applies to HAI through Bennett (2021). According to Bennett an agent needs to show coherent emergent behavior to create an intentional stance instead of using human-like surface features[8].

The system interacts through emails which contain only textual information without any human-like presentation. The agent demonstrates its personality and intelligence through three main elements which include:

- Linguistic style

- Clarity and relevance of content
- Proactivity (e.g., suggesting alternatives, asking clarifying questions)

The system's ability to generate intelligent responses through text-based communication helps users believe the agent possesses intelligence. The system generates intelligent responses through its assistant-like communication style which provides detailed explanations and state summaries and proactive recommendation services. The system avoids visual representation which prevents users from experiencing the "uncanny valley" effect. The system design focuses on meaningful text-based communication because Bennett advocates for agents that show real understanding instead of fake social signals.

The MaSMT integration through email functions as a purposeful assistant to generate helpful responses which motivates users to view the system as a collaborative tool instead of a black-box system.

F. MaSMT-Based Applications and the Need for Human-Friendly Channels

The MaSMT framework has proven successful in multiple real-world applications which show its ability to work in different environments. The framework has been used in three main applications:

- Octopus represents a Sinhala–English multi-agent chatbot system which uses MaSMT to process natural language queries through a GUI interface with multiple dedicated sub-agents for GUI interaction and NLP and search and learning functions[9].
- The MaSMT-based platform Rice Express operates in the agricultural sector to link rice farmers with buyers and transporters which enhances supply chain efficiency and shortens delivery times[10].
- The communication platform AgriCom enables MaSMT agents to link farmers with buyers and agricultural experts for better decision support and information exchange[11].

The modular design of MaSMT with its message-driven architecture enables domain-specific adaptations according to these systems. The current applications use two main interaction methods which include agent-to-agent communication and human-to-agent interaction through customized interfaces that need some technical knowledge. The implementation of AgriCom in rural areas faced challenges because users needed to use specialized software yet many potential users lacked digital skills and proper equipment.

Our research focuses on using email as the interaction platform because this tool provides universal understanding and widespread availability to make human participation more accessible. Users can access the MaSMT ecosystem through basic phone or computer email functionality without needing to learn new software. The transition from GUI-based to email-based interaction represents a fundamental advancement for making MaSMT-based systems accessible to all users.

G. Research Gap and Positioning of Email-Based MaSMT

The current literature lacks any study which implements email as a built-in communication system for multi-agent systems to handle Human-AI Interaction. The research introduces a new method which unites human-in-the-loop control with adjustable autonomy through email communication. The system operates through an email-based communication system which provides asynchronous access to all users. Jiang et al. (2024) state that upcoming HAI research needs to develop AI systems which provide equal access to all users regardless of their technical abilities[12]. The system enables basic email users to control complex agent systems through email interactions without requiring app installation or specialized syntax knowledge or stable real-time connectivity.

The introduction of this new HAI channel creates multiple operational difficulties for users.

- Email messages contain unclear language that users need to interpret.
- The system requires users to wait for extended periods before receiving feedback while maintaining continuous interaction.
- The system needs to preserve information throughout multiple email exchanges that span different threads.
- The system faces security threats because users receive spam messages and fake emails that contain sensitive information.

The system faces two major problems which match existing conversational agent research challenges: users fail to understand each other's intentions and agents experience communication failures. The system requires advanced natural language processing capabilities together with clarification methods which use follow-up emails to verify unclear requests[8]. The system requires authentication systems and validation protocols to prevent unauthorized users from executing essential system commands.

The proposed email-based MaSMT integration provides an advanced user interface which extends beyond standard agent-to-agent and GUI-based MaSMT systems. The system provides more reliable and flexible intelligent systems through its direct integration of human communication within the MAS framework.

Table I: The HAI Knowledge Map

Topic / Theme	**Main Insight**
Adjustable Autonomy in HAI	Agents dynamically shift autonomy levels, allowing human intervention in critical situations.
Conversational Agents & Deep NLP	Modern chatbots use deep learning but assume real-time interaction and lack long-term context.
Asynchronous HAI	Asynchronous interaction is under-investigated; most work focuses on synchronous modalities.
Scenario-Based Frameworks (FSDL, scripts)	Visual scripting simplifies known scenarios but agents struggle outside scripted behaviors.
Engagement in HAI	Engagement models rely on real-time multimodal cues and continuous interaction.
Intentionality & Intentional Stance	Users attribute minds to agents showing coherent, meaningful behavior over mere appearance.
MaSMT Applications (Octopus, Rice, AgriCom)	MaSMT is flexible across domains but typically accessed via custom interfaces by skilled users.
Inclusivity & Democratic AI	HAI should be accessible to users with minimal technical skills and limited infrastructure.
Breakdown Recovery & Security	Misunderstanding and breakdowns require explicit recovery strategies and robust safeguards.

III. Research Elaborations

A. Email-Integrated MaSMT Architecture

The system architecture “Fig. 2” builds upon MaSMT core framework through an email-based communication system which enables human users to interact with the agent society. The core component of MaSMT multi-agent platform consists of one Manager Agent and multiple Ordinary Agents that form swarms and maintain internal messaging infrastructure through global and local message queues[13]. The Email Gateway Agent (Email Interface Module) serves as a connection point between external email servers and the MaSMT messaging system. The gateway agent performs two essential functions which include email processing for incoming messages and email generation for agent-produced content.

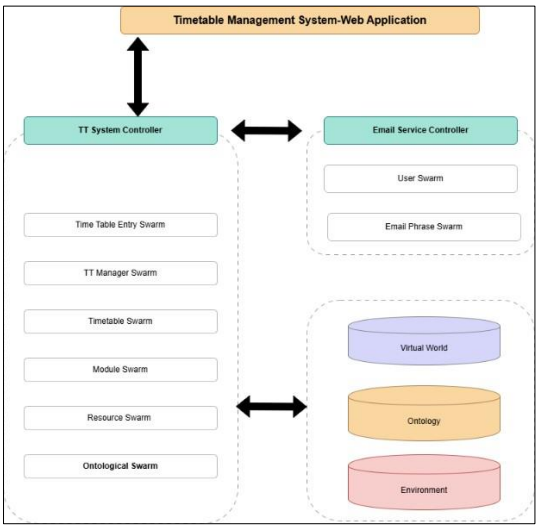

Figure 2: System Architecture of the Timetable Management System

The MaSMT system operates through message-based communication “Fig. 3” where agents send asynchronous messages that follow FIPA-ACL standards with performative and content elements with sender and receiver IDs and ontology references and additional metadata “Fig. 4”. The system Manager Agent functions as the central coordinator that directs all operations. The system Manager Agent tracks all incoming requests and inter-agent communications that require higher-level task management through the Global Message Queue. The Email Gateway Agent transforms incoming user emails into MaSMT messages which it adds to the global message queue after content extraction and human user identifier assignment. The Manager Agent checks the queue for messages before selecting a response method based on the message content. The manager agent will direct the reservation request to a Scheduling Agent which operates as an ordinary agent for room availability checks within its swarm. The manager agent uses the Local Message Space of its agent swarm to delegate tasks by adding messages to the local queue of the chosen ordinary agent for execution.

Date Field	Type	Description
sender	MaSMTAbstractAgent	Denotes the identity of the sender of the message
receiver	MaSMTAbstractAgent	Denotes the identity of the receiver of the message
replyTo	MaSMTAbstractAgent	The conversation is to be directed to the agent
message	String	A message should be subject to the message
content	String	The original content of the message
ontology	String	Ontological information
type	String	Type of the message
data	String	Some data (image, URL etc)
header	String	Directive field of the message (MaSMT send a message with considering it header
language	String	Language and encoding type
Conversation	int	Conversation ID of the message

Figure 3: MaSMT Message

Message Header	Description
Agents	The controller sends a message to a particular his client(s) who has given group and role
AgentGroup	The controller sends a message to a particular his client(s) who has given the group
AgentRole	The controller sends a message to a particular his client(s) who has given the role
Local	The agent or controller sends a message to it sawm who has given role and group
LocalRole	The agent or controller sends a message to it sawm who has given the role
RoleOrGroup	The agent or controller sends a message to it sawm who has given role or group
Broadcast	The agent or controller sends a message to it sawm
Root	The controller can send a message to Root agent
Controller	The agent can send a message to its Controller
NoticeBoard	The agent or controller sends a message to the noticeboard
MailMessage	The agent or controller send a message as a email message

Figure 4: MaSMT Message Parsing

The design of Ordinary Agents in MaSMT allows them to execute specific tasks which remain within their defined scope. Our system includes three ordinary agent types which are responsible for scheduling requests and database operations and natural language processing for email text analysis. The manager agent supervises these agents who retrieve messages from their local queues to perform their assigned tasks. The Scheduling Agent performs room reservation tasks after extracting email content details about time and date and room type to check available rooms through the Database Agent. The agent will create a response after completing its

task. The manager agent tracks all workflow activities and initiates communication between ordinary agents when necessary through local or global queue message transmission. The global message space enables Manager agents from different swarms to exchange messages with each other “Fig. 5”. The system design enables easy expansion to multiple manager agents who handle different email domains through their ability to share information when their work areas overlap.

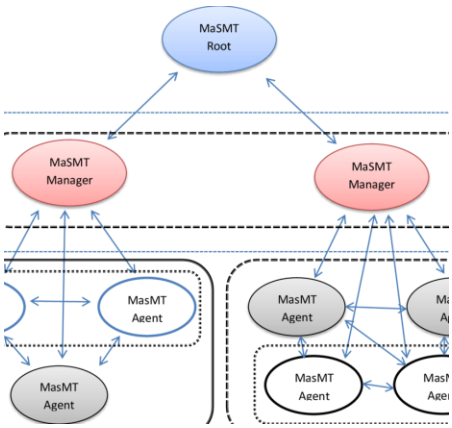

Figure 5: Hierarchical Structure of the MaSMT Multi-Agent System

The system uses email as its communication interface to transmit messages between users and agents. The manager agent directs the Email Gateway Agent to send messages to human users after result generation. The manager agent directs the Email Gateway Agent to send messages to human users through an Email Outbound Queue which the Email Interface monitors. The interface agent transforms message content into email format by assigning email body content and using subject lines to show response types such as "Reservation Confirmation" or "Query Clarification." The system uses standard email protocols (SMTP for sending and IMAP/POP3 for receiving) to connect with email servers which can be institutional mail servers or any email service supported by system configuration. The system benefits from established email infrastructure because it eliminates the requirement for developing special networking software. The system can operate from any location with network access to the mail server because email serves as a universal communication method that enables users can access through their email clients.The system enables human agents to function through a mechanism which treats each user email as if it were an agent message within the system. The system gives each human participant a distinct identifier which serves as their agent identifier in the MAS (their email address or a user ID mapping). The Email Gateway Agent includes this identifier as the sender information when it adds email-derived messages to the system. The system now shows an external agent entity (the human) which communicates through the message bus to other agents. The agents process messages without understanding their origin between human users and software agents because they receive messages with specific content. The system design enables humans to perform agent tasks through email commands which makes their participation indistinguishable from software agents “Fig. 6”. The system enables agents to start email contact with humans through automated messages which activate human participation in decision-making processes. The system operates under flexible autonomy because agents perform routine tasks independently but they send emails to humans for assistance with complex or high-level decisions.

The message format of MaSMT uses FIPA-ACL standards which include reply-to and conversation-id fields that we utilize to preserve email thread continuity. The system includes a reference header which email clients use to group messages and maintains a conversation-id for internal tracking of user responses to original context. The system maintains active email conversations through internal state management because users need to ask follow-up questions through new emails. The system needs to store information about user references because users need to access their previous messages during their current interaction.

The development of the email interface required JavaMail API standard libraries for gateway agent creation and a basic command system for email content processing. The system processes structured commands in email bodies through its current implementation which supports both key-value pairs and specific keywords (users can send "Reserve Room; Date: 2025-11-20; Time: 10:00-12:00; RoomType: Lecture Hall" in the email body). The system uses a structured format for message processing which makes it easier to understand and decreases the chances of misinterpretation. Users can enter their requests through free-form natural language when they want to make their interaction more human-like. The Email Gateway Agent uses a basic NLP module to extract user intentions from free-text requests before it fills in necessary reservation details. The system uses pattern-based interpretation and pre-trained language models to analyze user input according to Ahmed et al. (2024) in a restricted domain-specific context. The system accepts both structured and unstructured input to provide users with user-friendly access and reliable operation[5]. Users who prefer plain English can use it for simplicity but advanced users and system integrations can use structured commands for exact control.

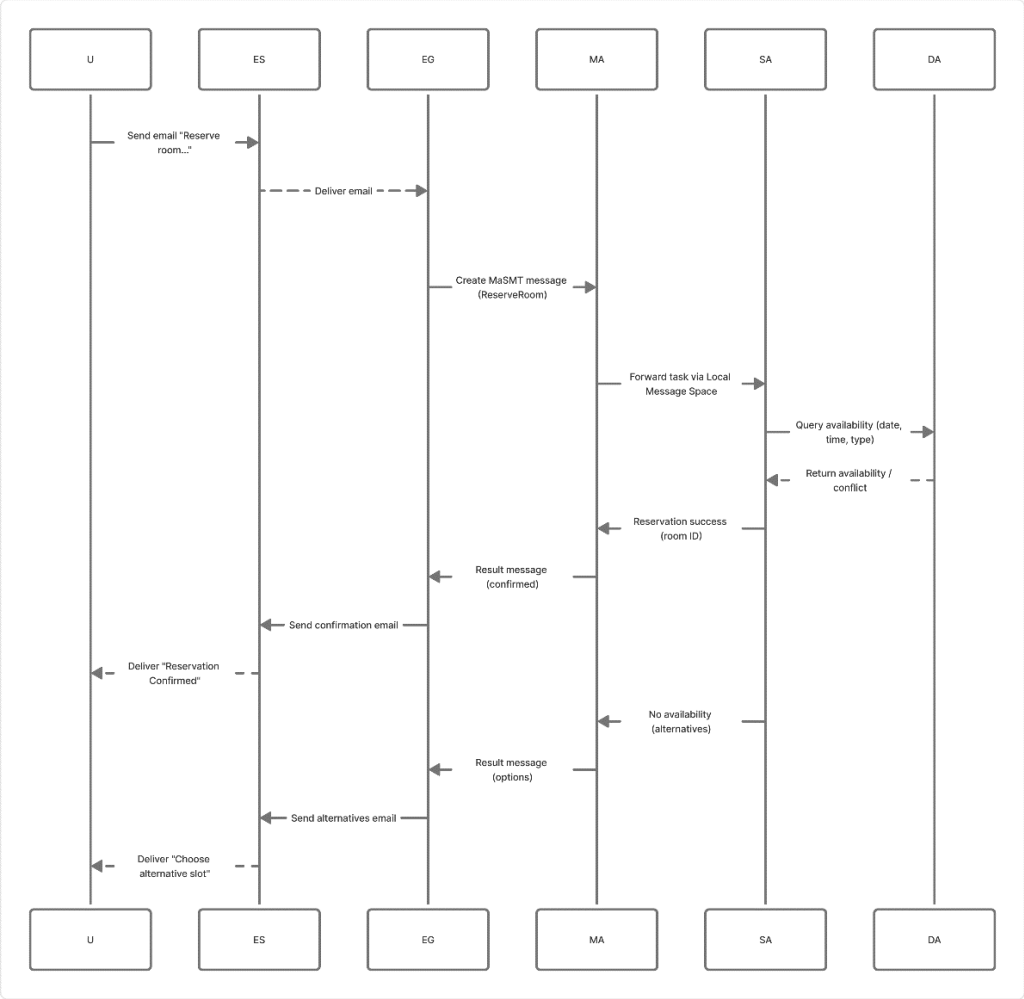

Figure 6: Human-to-MAS Interaction Flow for Room Reservation via Email

The system processes email requests through its multi-agent system (MAS) using the same procedures that apply to software agent requests. The system achieved this benefit because we did not need to modify the internal framework of MaSMT to support human participation. The email agent functions as an adapter layer which converts email data into agent-compatible information. The system operates through its built-in message handling system which performs task scheduling and execution and message queuing without requiring any changes to its existing functionality. The system maintains all its operational capabilities because it operates as a distributed system with peer-to-peer modules between manager agents. The system maintains its ability to scale up when more users join the system. The system allows multiple email gateway instances to operate independently for different departments and roles which then send messages to manager agents through the global message space. The system's performance depends on two factors: email server processing capacity and agent platform execution speed. The system operates at high performance because email servers process thousands of messages while MaSMT uses lightweight threading to enable multiple agents to run simultaneously.

B. Interaction Workflow and HAI Considerations

The following example demonstrates the system operation through a standard workflow of email-based MaSMT system interactions.

1. **User Inquiry:** A human user creates an email containing their task requirements or questions before sending it to the system email address (e.g. agent@domain.com). A lecturer would write: "Subject: Classroom Booking Request; Body: I need a computer lab for a guest lecture on Nov 20 from 10-12. Approximately 30 students will attend."

2. **Email Reception and Parsing:** The Email Gateway Agent checks the mailbox through IMAP at scheduled intervals. The system retrieves new messages while extracting essential information including subject lines and body content and sender addresses and timestamp data. The agent uses basic rules together with Natural Language Processing to extract user requirements and essential information from the email content. The system identifies this message as a room reservation request while extracting the following

information: date = Nov 20, time = 10:00–12:00, type = computer lab, size ≈30 people. The system generates a MaSMT message object which contains Message(type="ReserveRoom", date="2025-11-20", start="10:00", end="12:00", roomType="Lab", capacity="30", sender="user123"). The system uses user123 as a user ID to represent the lecturer who sent the email.

3. **Message Injection:** The gateway agent adds this message to the Global Message Queue where the Manager Agent can detect it (the system uses an alternative method to send messages directly to the Manager Agent's local queue when configured properly). The Manager Agent detects the "ReserveRoom" request message from the global queue which it monitors continuously.

4. **Task Delegation**: The Manager Agent starts tracking the request through logging before it assigns the work to its subordinate agents. The Manager Agent directs the message to a Scheduling Agent who operates within its team. The Scheduling Agent starts processing the received task after receiving it through its local message queue. The agent needs to request help from other agents when dealing with complex tasks that require database queries for room schedules or when handling recurring bookings with special requirements. The agents maintain communication through message exchange where the Scheduling Agent requests room availability from Database Agent.
5. **Agent Collaboration:** The system shows availability of computer labs during the requested time period. The Database Agent provides a list of rooms which match all specified requirements. The Scheduling Agent selects an appropriate room by using a strategy to find the smallest suitable lab which maximizes available space. The system would either automatically handle conflicts by following rules (e.g. first-come-first-served) or request human intervention through an email that asks users to select an alternative time or confirm an exception (this would generate a new outgoing message at this point to show adjustable system autonomy). The system operates without any scheduling conflicts at present.

6. **Response Preparation:** The Manager Agent receives the scheduling agent's successful reservation outcome which includes the assigned room information. The manager or scheduling agent creates a response message which will be sent to the user. The system generates a response message that states "Your reservation is confirmed. Room: Lab 202, Date: Nov 20, Time: 10:00-12:00." The agent includes booking identification numbers and cancellation links in the message. The system adds a sender identifier "user123" to the message which targets the original requestor.

7. **Email Sending:** The Email Gateway Agent checks the outbound message queue to find the ready-to-send response. The system transforms the message into an email format by adding "Re:" to the subject line and creating a respectful email body that contains all confirmation information. The system uses SMTP to deliver the email to the user's address which it obtained through message parsing or user ID mapping.

8. **User Receives Reply:** The human user finds their requested information or confirmation in an email from the system which appears in their inbox. Users who need additional interaction can answer the email directly. Users who entered incorrect dates through mistake can send a reply to request date changes for their reservation. The system processes this reply through Email Agent parsing which detects the request as a reservation update (this triggers a different workflow for updating existing bookings).

The workflow system implements multiple HAI principles which address user needs.

The system delivers feedback through email messages which include confirmation messages and error notifications. The agent sends an email to users when their requests cannot be fulfilled because no rooms are available. The system maintains user involvement through email communication which operates similarly to real-time conversational agents. Users maintain flexibility to respond at their preferred time which extends their interaction duration. The system enables users to maintain continuous interaction through considered responses which reduces their immediate engagement level.

The system uses natural language communication through complete sentences and polite language to create an impression of purposeful service delivery instead of automated software responses. The system email states "Our system located an available room which we reserved Lab 202 for your use. Please let me know if you require any modifications to your booking." The system presents itself as an autonomous assistant which understands user needs and actively works to achieve satisfaction through its wording. Research from HAI about trust development and mental model understanding shows that users build trust when agents provide explanations and confirmations. The email format allows users to receive detailed information because they expect complete content from their email messages. The email thread contains all exchanged messages which become part of the permanent record. The built-in logging system provides users and system administrators with access to all instructions and agent responses which creates an environment of trust and accountability. The system fulfills requirements for human-agent cooperation transparency which Ramchurn et al. (2016) established as essential for agent behavior trust and clarity[14].

The system architecture functions independently from classroom reservation tasks. The system design allows flexible parsing and message-driven operations which enable it to process different scenarios. The system would process requests from farmers and traders through email messages for crop status reports and transport requests in the same way as our scheduling example. The system requires

only two changes to its parsing system for domain-specific language processing and the addition of new ordinary agents for specific tasks such as Logistics Agent for truck scheduling. The system allows users to add new scenario templates through regex rules or classifier training for request type detection. The system will develop its ability to understand different languages through email processing which can be achieved by implementing online learning or model fine-tuning (although this feature is not present in the current version). The system demonstrates flexibility in handling complex tasks which match the requirements of real-world environments.

The system includes fundamental security measures which form part of its operational framework. The Email Gateway Agent allows users to specify which domains and users can send emails to the system through its configuration options. The system should either discard emails from unknown senders or direct them to human administrators for authentication verification to prevent unauthorized access. Users must verify their actions through verification codes and additional confirmation steps for accessing sensitive functions. The system requires users to verify their actions through a confirmation code which they must enter after receiving it via email for high-impact operations such as financial transaction approval. The system requires users to verify their actions through a confirmation code which they receive via email before proceeding with the operation. The system implements these security checks to protect against email spoofing and system compromise. The system maintains email interaction logs for auditing purposes while it plans to use email content encryption when privacy protection becomes essential. The security measures protect the HAI email integration from becoming a security risk for the MAS.

IV. Results and Discussion

A. Use Case: Email-Driven Classroom Reservation System

The email-based MaSMT integration practicality becomes evident through our implementation of a Classroom Reservation System as a case study. The system operated in a university department environment to help users reserve classrooms and laboratories for their lectures and meetings and events. The system selection focused on this particular task because it represents a typical departmental requirement which requires team collaboration and conflict management and regular human-machine dialogues through web-based forms and email correspondence. The system demonstrates human-agent interaction patterns through its implementation of email-based automation for typical daily tasks.

System Setup: The MaSMT manager agent operated as the deployment manager which controlled three ordinary agents ScheduleAgent and DatabaseAgent and NotifyAgent that ran as Java threads under the MaSMT framework. The ScheduleAgent contained the functionality to verify availability and select suitable rooms; DatabaseAgent operated as a basic database system which stored classroom schedules and room capacities; NotifyAgent functioned as an email template creator which generated user-friendly email responses. The NotifyAgent function could be integrated with the Email Gateway but we maintained separate entities to preserve the gateway's protocol handling capabilities from its content creation responsibilities. The Email Gateway Agent operated through an institutional SMTP/IMAP server which checked messages from a specific email address (e.g. classroom-booking@univ.edu). The system kept an authorized user email database which included faculty and staff members to automatically send rejection emails with explanation to unauthorized senders.

Scenario Execution: The system received email requests from multiple faculty members and administrative staff during a two-week trial period for their room booking requirements. The system provided users with basic instructions which included the email address to use and examples of acceptable request formats. Users sent their requests through complete sentences because they treated the system as if it were a human secretary. The system processed the following email interactions from users:

- User A: “I need to reserve a lecture hall for my class on December 1 between 10:00 and 12:00. The class contains 50 students. I appreciate your help.”

The system sent an automatic response to the user which stated “Hello! Your request has been received. The system checks for available lecture hall space during December 1 from 10:00 until 12:00 for 50 students.” (The system used NotifyAgent to generate this immediate response which confirmed email processing had started. The system sent a second email containing the booking outcome two minutes after receiving the initial request.)
The system sent a confirmation email to the user which stated that Room 301 (Lecture Hall) was reserved for December 1 from 10:00 until 12:00 for 50 people. The system assigned reservation ID #AHJ-59 to this booking. The system will stop responding to this email thread because it detects that the conversation has reached its end.

- User B: “I need to book a computer lab for November 20 between 2:00 PM and 4:00 PM because I am hosting a workshop which requires 30 participants.”

The system sent an apology message because all computer labs were fully booked during November 20 between 2:00 PM and 4:00 PM. The system found two available time slots for you which include Lab 5 from 4:00 until 6:00 PM on November 20 and Lab 3 from 2:00 until 4:00 PM on November 21. Please select one of these available time slots for your needs.

The system provided two possible time slots for the user to select from. The user selected one of the available time slots.
User B: “Please reserve Lab 3 for November 21 between 2:00 PM and 4:00 PM.”
The system confirmed your request by stating that Lab 3 is reserved for your workshop on November 21 from 2:00 until 4:00 PM. The system assigned reservation ID #BKD-77 to this booking. The booking service appreciates your use of our system.

The agent demonstrated its ability to manage email-based dialogues through these two examples which showed its natural interaction capabilities. The agent needed to negotiate with the user because it could not meet their request so it presented two possible solutions. The system asked the user to decide between two available options instead of taking control of the situation or simply rejecting the request. The email-based communication system operated without any issues because the user needed thirty minutes to respond while the system operated at the same pace. The system operated with flexible autonomy because it stored the task information and restarted its work when the human provided input according to Mostafa et al.'s agent behavior model.

User Experience and Feedback: The participants provided informal feedback through a survey which collected their responses. The system proved useful to users because it operated through their regular email system without requiring them to access a separate portal or learn new interface rules. A faculty member expressed his satisfaction with the system by stating that email remains his primary choice for scheduling purposes. The system operated like an assistant who provided fast responses through email. Users could initiate requests through their mobile devices while moving from one location to another. The system benefits from using existing user behaviors and devices according to this feedback. The user appreciated that the agent responded with complete sentences instead of brief confirmations because it created a professional and polite atmosphere. The system's well-designed email responses created an impression of agent intelligence and attention according to Bennett (2021) because authentic helpful behavior makes users believe they interact with an intelligent system[8].

The system received negative feedback from users who tried to send complex or ambiguous requests that the system failed to understand. A user sent an email requesting either a seminar room or two adjacent small classrooms for a joint session during next Monday afternoon. The request contained multiple possible interpretations together with specific conditions. The Email Agent failed to understand the request so it booked one small classroom before sending a confirmation message. The user expected the system to ask for clarification because they listed multiple possible solutions. The current NLP system lacks ability to handle complex human requests because it needs dialogues to clarify meanings which we only implemented for basic conflicts. The system requires an improved method to handle situations when it remains uncertain according to Benner et al. The system needs an improved method to handle situations when it remains uncertain. The system will implement advanced dialogue management to handle complex requests through email by asking users to confirm their intended meaning when parsing confidence levels are low or multiple interpretations exist. The system should ask users to choose between one seminar room and two adjacent small rooms for the next Monday afternoon. The system should ask users to select their preferred option through email so it can proceed with assistance. The system would achieve better user communication through the addition of these features which Tolzin & Janson (2023) identify as essential for conversational agents[15].

The email-based system required additional time for processing compared to web form booking but it maintained acceptable performance for its intended purpose. The complete process from user email submission to receiving confirmation took between 1 to 2 minutes when users factored in their reading and response times. The system processed emails to our server within less than 5 seconds while agent processing took under 1 second because the tasks involved minimal computation. The system needed to wait at least one minute before sending the response email because email servers operated at their own pace and performed spam checks. The booking process operates at an acceptable speed because users do not need to make urgent requests. The automated system processed requests at a speed that surpassed the response time of some human staff members according to one user.

The system processed 50 email exchanges during the trial period which included multiple messages in single email threads. The system processed a small number of requests. The email polling rate and email processing speed would become the primary performance limitations when handling large volumes of requests exceeding hundreds or thousands per day. The system can achieve better performance through two methods: increasing the gateway agent's mail check frequency and using multiple threads to send multiple emails simultaneously. The MAS system enables agent creation for handling multiple requests through its manager agent which can establish new SchedulingAgent instances for each request. The EnSiMaS translation system operated with 100 agents in parallel according to MaSMT scalability demonstrations in other domains. The processing of emails might create performance issues when NLP operations become complex but implementing message queuing and request prioritization can solve this problem[16]. The system maintains separate processing and communication channels through message queues which prevents delays in one system from causing the other system to fail. The system maintains operational continuity because email processing delays do not affect agent computation or vice versa.

B. Impact on Human-Agent Interaction

The implementation of email functionality within a multi-agent system creates new positive dynamics for Human-Agent Interaction (HAI).

The system proved accessible to users who received minimal training according to our research findings. The research by Jiang et al. supports user-focused AI development methods[12]. The system becomes more accessible through email communication because users can interact with it using everyday language. The system accepts requests from users who lack technical expertise because they can send emails to initiate interactions. The agent system becomes more inclusive because it accepts users from all backgrounds through its designed functionality. The system will gain more diverse user data and feedback through its inclusive design which will enhance its performance and fairness.

Humanization of Agents: The email communication system enabled agents to present themselves as human entities to users. Users began to treat the system as if it were a person by using personal greetings (e.g. "Hi, can you please... Thanks!") and the system's courteous answers strengthened this perception. Users recognized the automated nature of the system yet the communication format followed human-like patterns. Users tend to express their requirements through detailed explanations when interacting with automated systems because they would do so when speaking to a human being. The system's use of an AI requires clear disclosure through its email signature to prevent users from misunderstanding its nature. The "Classroom Booking Assistant" email signature did not reveal that the system operated through artificial intelligence. The system will add clear statements to its emails because this approach follows HAI best practices for maintaining proper user trust and expectation management.

The system requires users to maintain their willingness to interact through the interface at specific times. The trial results demonstrated that users would continue using the system after establishing trust because they showed repeated usage from faculty members who had positive initial experiences. The system received continuous participation through CC functionality from users who involved it in their thread discussions. A staff member sent an email to another person about event planning while including the booking agent in the CC field to prevent date conflicts from occurring. The agent processed the CC'd email because it received the message as one of its recipients and then sent a response containing event date availability to all parties involved. The agent joined an unexpected human email exchange through its participation in the conversation. The agent received such high acceptance from users because they used it as their collaborative work tool. The agent participated in multiple email conversations which required it to recognize information that was not explicitly directed toward it. The system operated successfully with basic cases although it lacked complete preparation for this functionality. The agent would become more effective as an AI that users could include in their email threads to handle all mentioned logistical tasks.

User Survey Results Show System Trust Levels Were High for a New Technology. Users trusted the system because they could view all email correspondence and because the agent provided multiple choices instead of making automatic decisions. The user stated: “The system maintains my control because it requests my input when it faces confusion. The email records serve as evidence which proves all decision-making processes.” The system design enables users to view the agent as a helpful tool which executes their instructions instead of behaving like an unmanageable black-box. The system maintained user involvement during all important decision-making processes to prevent agent actions from contradicting user preferences. Users rarely needed to understand agent decisions because the correspondence provided clear explanations. The system follows explainable AI principles and HAI design standards which demonstrate agent behavior to users for building trust[17].

Adaptability and Learning: The email interface enables the system to process various requests because backend agents support these functions. The system received a new Equipment Booking agent for projector and laptop management because users began requesting this service during the middle of the trial. The Email Gateway parser received updates to detect "projector" and "laptop" keywords while the system added an Equipment Agent to handle equipment requests. The system processed equipment requests through its existing email interface after users started sending requests to the same address for equipment availability information. The system demonstrates scenario adaptability through its ability to implement new features quickly. The system operates through email interfaces which stay unchanged while agents receive expanded capabilities and understanding. Users attempted scenarios which extended past our defined boundaries to demonstrate how the system would adapt to new situations through user-driven innovation. The system demonstrates its ability to grow through user-initiated attempts even though not all requests resulted in successful outcomes. The system demonstrates its ability to learn from user interactions through time-based accumulation of understood commands and conversation patterns. The system can learn from user interactions through two methods: machine learning analysis of email logs or developer maintenance of its knowledge base. The system develops into an adaptive agent which enhances its performance through continuous operation.

The research shows promising results but researchers encountered multiple obstacles during their study.

- **Natural Language Understanding:** The flexible nature of email communication allows users to express themselves through different sentence patterns and multiple requests within one message. The rule-based parsing system failed to process complex requests that appeared in emails with flexible language. A better NLP system which uses large language models

would improve understanding but it would require additional computational resources and precise model validation to prevent wrong system responses. The agent needs to achieve 100% human intent understanding but researchers should focus on enhancing this capability for successful deployment[18]. The system could use user profile information and previous system interactions to clarify unclear messages by learning specific user terminology such as "the lab" referring to a particular room.

- **Latency and Asynchrony:** Users experienced delays in most cases but some users attempted to achieve near-synchronous communication through sending multiple emails quickly for immediate responses. Users who use email as a chat platform will experience slow performance from the system. The system needs to recognize when multiple emails arrive quickly because it indicates an urgent matter so it should activate either real-time functionality or human operator notification[19]. The system should integrate instant messaging for users who need immediate responses but this would depart from the fundamental email-based system. Email systems work best for non-urgent negotiations when users understand they cannot expect immediate responses. The method proves unsuitable for critical situations which need immediate responses because it introduces delays in communication.

- **User Training and Errors:** A few users experienced formatting problems when they sent emails containing Excel files because they expected agents to extract information from these files but the system failed to do so. Users need to understand both the system's operational range and its operational boundaries to use it correctly. Our trial participants needed time to learn system operations because they discovered that the agent processed only the first request when users included multiple requests in a single email due to our parser restrictions. We distributed a one-page guide to users during the first week of the trial to explain system operation details[20]. The deployment of user onboarding procedures at any level would prevent users from becoming frustrated with the system. The system should preserve its natural interface because users should not need to use a structured email form.

- **Evaluation Metrics:** The project will use quantitative methods to assess HAI improvements through future research. The system's adoption rates and user retention serve as indicators for measuring engagement. The system achieves successful task outcomes in 94% of email conversations according to our trial results. The system needs human intervention for 3 out of 50 exchanges because users fail to understand system instructions or make unsupported requests. The system needs additional data to determine if its performance will enhance through learning processes[21]. User satisfaction assessment will occur through survey responses and monitoring user behavior to detect any return to previous methods. The current satisfaction data shows positive results but additional information would confirm this finding.

- **Interface Evaluation Through Comparison with Other Systems:** The evaluation of our email interface requires analysis against other HAI systems. The email interface provides unstructured input but enables users to access it from anywhere while offering more flexibility than web applications. The email interface provides better support for detailed information transfer through text because it allows users to send dates and lists easily. The email interface provides better accessibility than voice assistants because it does not require immediate responses and it maintains a record of all interactions. The system provides equal functionality in both noisy and silent environments because it does not depend on speech recognition technology. The evaluation process requires assessment of email interface advantages and disadvantages to determine its suitability for specific multi-agent applications[18]. The system proves effective for tasks which require asynchronous operations including scheduling and information requests and collaborative planning activities. The system requires a combination of SMS and push notification for urgent alerts but uses email for complete follow-up communication.

The classroom reservation case study proves that email-based communication enables human participation in multi-agent systems which results in more human-like and efficient Human-Agent Interaction (HAI). The agents processed actual user requests and maintained natural conversations while users interacted with the system as if it were operated by a human which confirms the method's user-friendly nature. We will present the main findings of this research along with possible directions for future development and expansion.

V. Conclusion

The research established a new method to integrate email communication with MaSMT multi-agent systems which improves Human-Agent Interaction capabilities. The research started by showing how traditional MAS systems create closed environments which prevent humans from directly interacting with them. The research showed how email functions as an accessible asynchronous communication tool which enables humans to join multi-agent systems through command issuance and agent dialogue. The research methodology used email communication to support HAI development through its alignment with adjustable autonomy trends and its solution of asynchronous interaction gaps and its consideration of user engagement and intentionality factors for AI system acceptance.

The Email Gateway Agent of the email-augmented MaSMT system connects the MAS to standard email servers through an interface which translates human-friendly emails into agent-readable messages. The system maintains its core design structure from MaSMT while introducing a new communication interface that expands agent accessibility. The email-based classroom reservation system served as an actual implementation example which demonstrated the system's operational success and user satisfaction with its user-friendly design. The system allowed users to start multiple conversations with agents which resulted in successful task accomplishment through email-based interactions with intelligent agents.

Benefits of the email-based HAI model provides multiple distinct advantages to users. The system provides agent-based services through email access which eliminates the requirement for users to learn complex interfaces or receive specialized training[22]. The system enables agents to request human assistance during specific situations which enhances their ability to make better decisions when algorithms prove insufficient (e.g. handling schedule conflicts through user-defined priority). Users and agents benefit from asynchronous communication because they can create well-thought-outlined messages which results in better decision quality. Users maintain trust in the system because they can access email threads which show all agent actions performed. The system development process benefits from email communication because it eliminates the need to build delivery systems and notification systems and storage facilities which already exist in reliable services thus enabling researchers to concentrate on agent logic development.

The system requires additional research to address its current limitations. The system requires better natural language processing capabilities to handle open email formats because its current solution needs improvement. The system will use Transformer-based language models for flexible request parsing but will maintain a deterministic layer for essential interpretation tasks to prevent errors. The system needs to develop capabilities which enable users to send different types of content including forms and photos for agents to process. The system needs to integrate computer vision and OCR agents into MaSMT to extract information from user input which requires new HAI solutions for agent summary generation.

The system will receive direction from real-world deployment requirements which will determine its future development path. A university-wide or enterprise deployment of this system would require email gateway distribution or cloud-based email processing functions to handle simultaneous threads from hundreds of users. The user experience will suffer when email responses become delayed during periods of high system usage. The system can prevent user experience degradation through request prioritization and department-specific email addresses. The system includes a human operator fallback mechanism which transfers threads to human secretaries or helpdesk staff when agents fail to resolve requests through multiple exchanges. The hybrid intelligence model achieves its goal through email-based handoffs because users can respond to existing threads which agents create.

The research field of Human-Artificial Intelligence (HAI) needs to answer questions about how users behave when they interact with AI systems through asynchronous communication channels. The absence of immediate feedback through email might create advantages in particular situations because it provides users with time for reflection[23]. Users tend to personify email agents differently than they do voice or chatbot agents. Users tend to view email agents as professional tools because they use email for official business communications which creates both advantages and restrictions. Research into user perceptions will enable developers to create agent personalities which match user expectations and follow established domain standards.

Research should focus on understanding how users interact with the system throughout their extended usage period. Users will maintain their system usage when they establish regular habits with the system but they might discover unanticipated usage methods (similar to CC'ing the agent). Research conducted over time will help scientists understand how users adapt their HAI email interactions through experience. Users tend to delegate more tasks to the agent when they feel comfortable with the system because they treat it as an autonomous assistant. The agent needs to demonstrate its operational boundaries and success indicators through adaptive intentionality calibration to support user trust development and autonomy growth.

The research demonstrates that email functions as an effective method to establish human-agent interaction within MaSMT. The research expands HAI capabilities by introducing asynchronous text-based teamwork which represents a new exploration area in human-agent interaction. The positive user feedback from our case study demonstrates that this model should be tested in additional fields. The system becomes more accessible to users because it operates through their familiar daily communication channels. Intelligent systems achieve better alignment with human values through this approach because people naturally send emails to request assistance from intelligent agents. The email-based peer-to-peer communication model between humans and agents shows potential for multiple applications including smart home management and business process automation. The development of human-agent ecosystems becomes possible through technological advancements which solve ambiguity and security issues to create seamless systems that enhance human capabilities through background artificial agent support accessible via email.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Department of Computer Engineering and the dedicated academic staff of the Faculty of Computing at General Sir John Kotelawala Defense University (KDU). Their unwavering support, expert guidance, and continuous encouragement have been invaluable throughout the course of this research. The knowledge and mentorship they provided played a pivotal role in bringing this project to successful completion.

REFERENCES

[1] B. Hettige, A. S. Karunananda, and G. Rzevski, "MaSMT: A Multi-agent System Development Framework for English-Sinhala Machine Translation," Int. J. Comput. Linguist. Nat. Lang. Process., vol. 2, no. 7, pp. 411–416, 2013.

[2] B. Hettige, A. Karunananda, and G. Rzevski, "Thinking Like Humans: A New Approach to Machine Translation," in Proc. SLAAI-ICAI 2018, CCIS 890, Springer, 2019, pp. 256–268.

[3] S. A. Mostafa et al., "A Flexible Human-Agent Interaction Model for Supervised Autonomous Systems," Proc. ISAMSR, 2016, pp. 106, doi: 10.1109/ISAMSR.2016.7750747.

[4] S. Kusal et al., "AI-Based Conversational Agents: A Scoping Review From Technologies to Future Directions," IEEE Access, vol. 10, pp. 92337–92356, 2022, doi: 10.1109/ACCESS.2022.3201144.

[5] N. Ahmed et al., "Deep Learning-Based Natural Language Processing in Human–Agent Interaction: Applications, Advancements and Challenges," Natural Language Processing Journal, vol. 9, 2024, Art. no. 100112, doi: 10.1016/j.nlp.2024.100112.

[6] K.-Y. Chin, J.-M. Lin, Z.-W. Hong, and A. J. Lin, "An Agent Scenario Mechanism Supporting Human/Agent Interaction," Proc. IAT, 2005, pp. 175–178, doi: 10.1109/IAT.2005.58.

[7] C. Oertel et al., "Engagement in Human-Agent Interaction: An Overview," Frontiers in Robotics and AI, vol. 7, Art. no. 92, 2020, doi: 10.3389/frobt.2020.00092.

[8] C. C. Bennett, "Evoking an Intentional Stance During Human-Agent Social Interaction: Appearances Can Be Deceiving," in Proc. 30th IEEE RO-MAN, 2021, pp. 362–367, doi: 10.1109/RO-MAN50785.2021.9515420.

[9] B. Hettige and A. S. Karunananda, "Octopus: A Multi-Agent Chatbot," in Proc. 8th Int. Res. Conf., KDU, Sri Lanka, Nov. 2015, pp. 41–42.

[10] M. A. S. T. Goonatilleke, M. W. G. Jayampath, and B. Hettige, "Rice Express: A Communication Platform for Rice Production Industry," in Proc. SLAAI-ICAI 2018, Springer, 2019, pp. 269–277.

[11] H. M. H. R. Jayarathna and B. Hettige, "AgriCom: A Communication Platform for Agriculture Sector," in Proc. 8th IEEE Int. Conf. on Industrial and Information Systems (ICIIS), 2013, pp. 439–444.

[12] T. Jiang et al., "Human-AI Interaction Research Agenda: A User-Centered Perspective," Data Inf. Manag., vol. 8, 2024, Art. no. 100078, doi: 10.1016/j.dim.2024.100078.

[13] T. Samaranayake and B. Hettige, "MaSMT Integration for English to Sinhala Machine Translation," Dept. of Computing, KDU, unpublished manuscript, 2023.

[14] S. D. Ramchurn, A. Rosenfeld, and J. E. Fischer, "Special Issue on Human-Agent Interaction," Autonomous Agents and Multi-Agent Systems, Jan. 2016, doi: 10.1007/s10458-015-9295-3.

[15] A. Tolzin and A. Janson, "Mechanisms of Common Ground in Human-Agent Interaction: A Systematic Review of Conversational Agent Research," Proc. 56th Hawaii Int. Conf. Syst. Sci., 2023, pp. 342–351.

[16] D. Benner et al., "What Do You Mean? A Review on Recovery Strategies to Overcome Conversational Breakdowns of Conversational Agents," Proc. ICIS 2021, pp. 1–14.

[17] J. M. Bradshaw, P. J. Feltovich, and M. Johnson, "Human-Agent Interaction," Florida Institute for Human and Machine Cognition, Jan. 2011. [Online]. Available: https://www.researchgate.net/publication/267819585.

[18] B. Hettige, A. S. Karunananda, and G. Rzevski, "Multi-agent System Technology for Morphological Analysis," Proc. 9th SLAAI Annual Sessions, 2012, pp. 1–7.

[19] P. V. Vezeteu and D. I. Năstac, "Simulating the Evolution of Infectious Agents Through Human Interaction," Proc. IEEE SIITME, 2020, pp. 1–4, doi: 10.1109/SIITME50350.2020.9292242.

[20] L. D. S. B. Weerasinghe et al., "Resource Sharing in Distributed Environment using Multi-agent Technology," Int. J. Comput. Appl., vol. 167, no. 5, pp. 28–33, June 2017.

[21] L. D. S. B. Weerasinghe et al., "ITray: Multi-agent Solution for LAN Based File Sharing," in Proc. 9th Int. Res. Conf., KDU, Sri Lanka, 2016, pp. 81–87.

[22] J. Huang et al., "Human Decision-Making Modeling and Cooperative Controller Design for Human–Agent Interaction Systems," IEEE Trans. Human-Machine Syst., vol. 52, no. 6, pp. 1122–1134, 2022, doi: 10.1109/THMS.2022.3185333.

[23] A. Hosseinpanah, N. C. Krämer, and C. Straßmann, "Empathy for Everyone? The Effect of Age When Evaluating a Virtual Agent," in Proc. HAI '18, ACM, 2018, pp. 184–190, doi: 10.1145/3284432.3284442.

AUTHORS

First Author – Praveena Madhubhani, Department of Computer Engineering, Faculty of Computing, General Sir John Kotelawala Defence University,39-bce-0023@kdu.ac.lk

Second Author – Dr. Budditha Hettige, Senior Lecturer, Department of Knowledge Engineering and Communication, Faculty of Computing, University of Sri Jayewardenepura, budditha@sjp.ac.lk